IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Cynthia M. Merkin		
Assignee:	Dell Products, L.P.		
Title:	Redundant Source Event Log		
Serial No.:	10/027,618	Filing Date:	October 22, 2001
Examiner:	Ji-Yong D. Chung	Group Art Unit:	2152
Docket No.:	DC-03124	Customer No.:	33438

Austin, Texas
May 14, 2008

Electronically Filed

APPEAL BRIEF UNDER 37 CFR § 41.37

Dear Sir:

Applicant submits this Appeal Brief pursuant to the Notice of Appeal and Pre-Appeal Request for Review filed in this case on August 21, 2006. A Notice of Panel Decision from Pre-Appeal Brief Review was mailed on March 14, 2008, resetting the time period for filing an Appeal Brief to April 14, 2006. Applicants respectfully request a one-month extension of time in which to file the Appeal Brief resetting the time period to May 14, 2008.

The fee for this Appeal Brief is being paid electronically via the USPTO EFS. The Board is also authorized to deduct any other amounts required for this appeal brief and to credit any amounts overpaid to Deposit Account No. 502264.

I. REAL PARTY IN INTEREST - 37 CFR § 41.37(c)(1)(i)

The real party in interest is the assignee, Dell Products L.P. as named in the caption above and as evidenced by the assignment set forth at Reel 012417, Frame 0196.

II. RELATED APPEALS AND INTERFERENCES - 37 CFR § 41.37(c)(1)(ii)

Based on information and belief, there are no appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

III. STATUS OF CLAIMS - 37 CFR § 41.37(c)(1)(iii)

Claims 1, 3 – 17, and 22 - 32 are pending in the application. Claims 1, 3 – 17, and 22 - 32 stand rejected. The rejection of claims 1, 3 – 17, and 22 - 32 is appealed. Appendix "A" contains the full set of pending claims.

IV. STATUS OF AMENDMENTS - 37 CFR § 41.37(c)(1)(iv)

No amendments after final have been requested or entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER - 37 CFR § 41.37(c)(1)(v)

The present invention, as set forth by independent claim 1, relates to a method of accessing event data describing a failure via a system management controller included in a computer system (see e.g., Page 12, Lines 1-4). The method includes configuring the system management controller to monitor a task of writing data to an event log (see e.g., Page 12, Lines 4-5), the task being executed by a Basic Input Output System (BIOS) program in response to the failure (see e.g., Page 12, Lines 5-8), monitoring the task for completion (see e.g., Page 12, Lines 9-11) and accessing the event data if the task fails to complete (see e.g., Page 12, Lines 21-23) and writing the event log via the system management controller in response to accessing the event data (see e.g., Page 13, Lines 18-25).

The present invention, as set forth by independent claim 14, relates to a method of accessing event data on a failure of a computer system (see e.g., Page 13, Lines 6-7). The method includes executing a BIOS program to access the event data in response to a first failure of the computer system (see e.g., Page 13, Lines 9-11), triggering a watchdog timer in a system management controller of the computer system, the watchdog timer being triggered substantially concurrent to the first failure (see e.g., Page 13, Lines 10-13), configuring the watchdog timer to allow the BIOS program to complete in absence of a second failure(see e.g., Page 13, Lines 15-21), determining whether the execution of the BIOS program caused the second failure, the second failure forcing the watchdog timer to expire (see e.g., Page 13, Lines 15-20), and the system management controller accessing the event data when the watchdog timer expires (see e.g., Page 13, Lines 18-21) and writing the event log via the system management controller in response to accessing the event data (see e.g., Page 6, Lines 3-24).

The present invention, as set forth by independent claim 17, relates to a computer system which includes a processor (see e.g., Page 6, Lines 3-24), a memory coupled to the processor (see e.g., Page 6, Lines 3-24), a BIOS program stored in the memory, the BIOS program being operable to write data to an event log in response to a critical event (see e.g., Page 13, Lines 9-11) and a system controller coupled to the memory and the processor (see e.g., Page 10, Lines 8-15) wherein the system controller is operable to receive an indication of the critical event (see e.g., Page 12, Lines 4-5) and upon receipt of the indication (see e.g., Page 12, Lines 5-8), initiate operation of a timer (see e.g., Page 12, Lines 5-8), and determine whether the BIOS program has written the data to the event log within a configurable period of time defined by the timer (see e.g., Page 13, Lines 18-25) and writing the data to the event log if the BIOS program has not written the data to the event log (see e.g., Page 13, Lines 18-25).

The present invention, as set forth by independent claim 18, relates to a computer system having a processor (see e.g., Page 6, Lines 3-24) and a system controller (see e.g., Page 10, Lines 8-15) and a method of responding to an event, the method includes issuing an interrupt to the processor in response to the event (see e.g., Page 10, Lines 8-15), detecting the interrupt at the system controller coupled to the processor (see e.g., Page 10, Lines 8-15), initiating a timer in the system controller upon detection of the interrupt (see e.g., Page 12, Lines 5-8), attempting to write data to an event log by executing a BIOS program (see e.g., Page 13, Lines 18-25), and the system controller determining whether the execution of the BIOS program resulted in writing data to the event log (see e.g., Page 10, Lines 16-20) and writing data to the event log via the system management controller if the execution of the BIOS program did not result in writing data to the event log (see e.g., Page 13, Lines 18-25).

The present invention, as set forth by independent claim 20, relates to a method for accessing and writing event data to a log on failure of a computer system including a system management controller (see e.g., Page 10, Lines 8-15), the method includes monitoring a task of writing data to an event log via the system management controller, the task being executed by a Basic Input Output System (BIOS) program in response to the failure (see e.g., Page 10, Lines 8-15), monitoring the task for completion to determine whether the BIOS program was able to complete writing the data to the event log (see e.g., Page 10, Lines 16-20), accessing the event

data if the task failed to complete (see e.g., Page 10, Lines 16-20) and writing the event log via the system management controller if the task failed to complete (see e.g., Page 13, Lines 18-25).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL - 37 CFR § 41.37(c)(1)(vi)

Whether Claims 1, 3 – 17, and 22 - 32 were properly rejected under 35 U.S.C. § 102(e) as being anticipated by Kosugi et al., U.S. Publication No. 20010044841 (Kosugi) is respectfully requested reviewed on appeal. Whether Claims 2, 18, 19 and 21 were properly rejected under 35 U.S.C. § 103(a) as being unpatentable over Kosugi et al., U.S. Publication No. 20010044841 (Kosugi), in view of Davis, U.S. Publication No. 20030070115 (Davis) is respectfully requested reviewed on appeal.

VII. ARGUMENT - 37 CFR § 41.37(c)(1)(vii)

Claims 1, 3 – 17, and 22 - 32 are Allowable Under 35 U.S.C. § 102(e) Over Kosugi et al., U.S. Publication No. 20010044841 (Kosugi).

The present invention generally relates to a method for accessing and writing event data to a log on failure of a computer system that is independent of the computer system's operating system. In the method, if it was determined that the BIOS program was not able to write the data to the event log, e.g., when the BIOS program encounters a subsequent failure such as an occurrence of a second critical event, then the system controller is configured to respond to the second failure by writing the data to the event log. The execution of the BIOS program may result in a second failure, thereby preventing the BIOS program from being completed.

Kosugi discloses a maintenance support apparatus and a maintenance method for automatically notifying an external remote maintenance system of the occurrence of a trouble at the time of starting a computer system until a system application is started. The maintenance support apparatus includes a server management support board and an integrated management panel which monitors a system. A start processing unit conducts start processing including a self diagnosis processing and a BOOT processing of an operating system and then starts an application when power of a computer system is turned on. A trouble notification unit controls the power of the computer system and integrally monitors a trouble of the start processing unit

and a trouble during system operation. The trouble notification unit, provided as a server management support board, acquires log information stored in the start processing unit and notifies an external remote maintenance system of the log information as well as an alarm message through a dedicated network interface when the trouble notification unit detects the trouble of the start processing unit (system down).

The examiner cited to the following portion of Kosugi when setting forth that Kosugi discloses configuring a system management controller to monitor a task of writing data to an event log:

> The server maintenance support board 36 notifies the ***external remote maintenance server 16*** of an alarm message indicating that system down occurs and the system log (BIOS log) at the time since the power of the system is turned on until the application is stared using a network interface, i.e., if the server maintenance support board 36 is notified by the IMP panel board 34 of the system down caused by the trouble which occurs during a self diagnosis processing (POST diagnosis processing), a BOOT processing and an application start processing conducted by the baseboard 30 by means of an electronic mail and an attachment file thereof. (Kosugi, Para. 29, emphasis added.)

Kosugi does not disclose monitoring for the failure of a task to complete where the monitoring is via a system management controller that is included within the computer system in which the failure occurs.

In response to Applicants' remarks, the Examiner set forth:

> The applicant has incorporated limitations related to a controller "coupled between a processor bus and a local bus," in the independent claims. Reciting passages in Kosugi (the passages which have been cited in the prior Office action) in Remarks, the applicant has highlighted the term "the remote maintenance server." The applicant argues that the system controller in the instant application is local. Apparently, the applicant is trying to distinguish the claimed subject matter based on the distinction between local and remote controllers.
>
> Based on the Remarks, it seems that the applicant's misunderstands the rejections based on Kosugi's system. As provided in the previous Office actions, the controller 88 in Fig. 2 was cited to be "system management controller." The controller 88 is *local* to the system that is performing the logging, as is the system controller in claim 1.
>
> Note that, Kosugi monitors any errors (which would presumably include the failure to complete a task, such as writing to a log). (Final Office Action, page 2.)

However, Applicant's position is that the system management controller is within the system for which the event data is being generated, not merely the system that is performing the logging. This feature which is claimed in each of the independent claims is not disclosed or suggested by Kosugi. Nor is this feature disclosed or suggested by any of the other references cited by the Examiner.

Davis discloses a mechanism that allows remote diagnosis and management of a computer system, regardless of whether the computer system is installed as a stand alone unit or coupled to a computer network.

More specifically, Kosugi and Davis, taken alone or in combination, do not teach or suggest a method of accessing event data describing a failure where the method includes providing a computer system with a system management controller coupled between a processor bus and a local bus, configuring the system management controller to monitor a task of writing data to an event log, the task being executed by a Basic Input Output System (BIOS) program in response to the failure, monitoring the task for completion and accessing the event data if the task fails to complete and writing the event log via the system management controller in response to accessing the event data, all as required by claim 1. Accordingly, claim 1 is allowable over Kosugi and Davis. Claims 2 - 13 depend from claim 1 and are allowable for at least this reason.

Kosugi and Davis, taken alone or in combination, do not teach or suggest a method of accessing event data on a failure of a computer system where the method includes executing a BIOS program to access the event data in response to a first failure of the computer system, triggering a watchdog timer in a system management controller of the computer system, the watchdog timer being triggered substantially concurrent to the first failure, the system management controller being coupled between a processor bus and a local bus, configuring the watchdog timer to allow the BIOS program to complete in absence of a second failure, determining whether the execution of the BIOS program caused the second failure, the second failure forcing the watchdog timer to expire, and the system management controller accessing the event data when the watchdog timer expires and writing the event log via the system management controller in response to accessing the event data, all as required by claim 14.

Accordingly, claim 14 is allowable over Kosugi and Davis. Claims 15 and 16 depend from claim 14 and are allowable for at least this reason.

Kosugi and Davis, taken alone or in combination, do not teach or suggest a *computer system which includes a system controller coupled between a processor bus and a local bus* where the system controller is operable to receive an indication of the critical event and upon receipt of the indication, initiate operation of a timer, and determine whether the BIOS program has written the data to the event log within a configurable period of time defined by the timer much less *write the data to the event log if the BIOS program has not written the data to the event log*, all as required by claim 17. Accordingly, claim 17 is allowable over Kosugi and Davis.

Kosugi and Davis, taken alone or in combination, do not teach or suggest a method of responding to an event in a computer system having a processor and *a system controller where the system controller is coupled between a processor bus and a local bus* where the method includes issuing an interrupt to the processor in response to the event, detecting the interrupt at the system controller coupled to the processor, initiating a timer in the system controller upon detection of the interrupt, attempting to write data to an event log by executing a BIOS program, and the system controller determining whether the execution of the BIOS program resulted in writing data to the event log and writing data to the event log via the system management controller if the execution of the BIOS program did not result in writing data to the event log, all as required by claim 18. Accordingly, claim 18 is allowable over Kosugi and Davis. Claim 19 depends from claim 18 and is allowable for at least this reason.

Kosugi and Davis, taken alone or in combination, do not teach or suggest a method for accessing and writing event data to a log on failure of a computer system including *a system management controller which is coupled between a processor bus and a local bus*, where the method includes monitoring a task of writing data to an event log via the system management controller, the task being executed by a Basic Input Output System (BIOS) program in response to the failure, monitoring the task for completion to determine whether the BIOS program was able to complete writing the data to the event log; accessing the event data if the task failed to complete and writing the event log via the system management controller if the task failed to

complete, all as required by claim 20. Accordingly, claim 20 is allowable over Kosugi and Davis. Claims 21 - 32 depend from claim 20 and are allowable for at least this reason.

Claims 2, 18, 19 and 21 are Allowable Under 35 U.S.C. § 103(a) Over Kosugi et al., U.S. Publication No. 20010044841 (Kosugi), in view of Davis, U.S. Publication No. 20030070115 (Davis).

The present invention, as set forth by claim 2, relates to the method of claim 1, wherein the failure generates a system management interrupt and the BIOS program is triggered in response to the system management interrupt.

Kosugi and Davis are discussed above.

Kosugi and Davis, taken alone or in combination, do not teach or suggest the method of claim 1, wherein the failure generates a system management interrupt and the BIOS program is triggered in response to the system management interrupt, all as required by Claim 2 and as substantially required by claims 18, 19 and 21. Accordingly, Claims 2, 18, 19 and 21 are allowable over Kosugi and Davis.

VIII. CLAIMS APPENDIX - 37 CFR § 41.37(c)(1)(viii)

A copy of the pending claims involved in the appeal is attached as Appendix A.

IX. EVIDENCE APPENDIX - 37 CFR § 41.37(c)(1)(ix)

None

X. RELATED PROCEEDINGS APPENDIX - 37 CFR § 41.37(c)(1)(x)

There are no related proceedings.

XI. CONCLUSION

For the reasons set forth above, Applicant respectfully submits that the rejection of pending Claims 1- 32 is unfounded, and requests that the rejection of claims 1- 32 be reversed.

I hereby certify that this correspondence is being electronically submitted to the COMMISSIONER FOR PATENTS via EFS on May 14, 2008.

/Stephen A. Terrile/

Attorney for Applicant(s)

Respectfully submitted,

/Stephen A. Terrile/

Stephen A. Terrile
Attorney for Applicant(s)
Reg. No. 32,946

CLAIMS APPENDIX "A" - 37 CFR § 41.37(c)(1)(viii)

1. In a system management controller included in a computer system, a method of accessing event data describing a failure, the method comprising:

configuring the system management controller to monitor a task of writing data to an event log, the task being executed by a Basic Input Output System (BIOS) program in response to the failure;

monitoring the task for completion;

accessing the event data if the task fails to complete; and,

writing the event log via the system management controller in response to accessing the event data.

2. The method of claim 1, wherein the failure generates a system management interrupt and the BIOS program is triggered in response to the system management interrupt.

3. The method of claim 1, wherein monitoring the task comprises:

setting a configurable time of a watchdog timer, the task being configured to access the event data, and write the data to the event log in response to the event data, the task being completed within the configurable time set in the watchdog timer;

receiving an indication from the BIOS program on completion of the task.

4. The method of claim 3, wherein the task fails to complete when the task fails to receive the indication from the BIOS program.

5. The method of claim 3, wherein receiving the indication from the BIOS program comprises resetting the configurable time in the watchdog timer.

6. The method of claim 1, wherein the event data is stored in a memory of the computer system by a controller device included in the computer system.

7. The method of claim 6, wherein the controller device is a memory controller.

8. The method of claim 6, wherein the controller device is an I/O controller.

9. The method of claim 1, wherein the system management controller accesses the event data over a system bus of the computer system.

10. The method of claim 9, wherein the system bus is a SMbus.

11. The method of claim 1 further comprising, the system management controller writing the event log in response to accessing the event data.

12. (Previously Presented) The method of claim 1, wherein writing the event log occurs over a system bus of the computer system.

13. The method of claim 12, wherein the system bus is a SMbus.

14. (Previously Presented) A method of accessing event data on a failure of a computer system, the method comprising:

executing a BIOS program to access the event data in response to a first failure of the computer system;

triggering a watchdog timer in a system management controller of the computer system, the watchdog timer being triggered substantially concurrent to the first failure;

configuring the watchdog timer to allow the BIOS program to complete in absence of a second failure;

determining whether the execution of the BIOS program caused the second failure, the second failure forcing the watchdog timer to expire; and

the system management controller accessing the event data when the watchdog timer expires; and,

writing the event log via the system management controller in response to accessing the event data.

15. The method of claim 14, wherein the second failure is substantially similar to the first failure.

16. The method of claim 14, wherein the second failure occurs while a processor included in the computer system operates in a SMM mode.

17. (Previously Presented) A computer system comprising:

a processor;

a memory coupled to the processor;

a BIOS program stored in the memory, the BIOS program being operable to write data to an event log in response to a critical event;

a system controller coupled to the memory and the processor, the system controller operable to:

receive an indication of the critical event;

upon receipt of the indication, initiate operation of a timer;

determine whether the BIOS program has written the data to the event log within a configurable period of time defined by the timer; and,

write the data to the event log if the BIOS program has not written the data to the event log.

18. (Previously Presented) In a computer system having a processor and a system controller, a method of responding to an event, the method comprising:

issuing an interrupt to the processor in response to the event;

detecting the interrupt at the system controller coupled to the processor;

initiating a timer in the system controller upon detection of the interrupt;

attempting to write data to an event log by executing a BIOS program;

the system controller determining whether the execution of the BIOS program resulted in writing data to the event log; and,

writing data to the event log via the system management controller if the execution of the BIOS program did not result in writing data to the event log.

19. The method of claim 18 further comprising:

if execution of the BIOS does not result in the writing of the data to the event log before expiration of a time period established by the timer, causing the system controller to respond to the event.

20. A method for accessing and writing event data to a log on failure of a computer system including a system management controller, the method comprising:

monitor a task of writing data to an event log via the system management controller, the task being executed by a Basic Input Output System (BIOS) program in response to the failure;

monitoring the task for completion to determine whether the BIOS program was able to complete writing the data to the event log;

accessing the event data if the task failed to complete; and,

writing the event log via the system management controller if the task failed to complete.

21. The method of claim 20, wherein if the task failed to complete, the system management controller generates a system management interrupt and the BIOS program is triggered in response to the system management interrupt.

22. The method of claim 20, wherein monitoring the task comprises:

setting a configurable time of a watchdog timer, the task being configured to access the event data, and write the data to the event log in response to the event data, the task being completed within the configurable time set in the watchdog timer;

receiving an indication from the BIOS program on completion of the task.

23. The method of claim 22, wherein the task fails to complete when the task fails to receive the indication from the BIOS program.

24. The method of claim 22, wherein receiving the indication from the BIOS program comprises resetting the configurable time in the watchdog timer.

25. The method of claim 20, wherein the event data is stored in a memory of the computer system by a controller device included in the computer system.

26. The method of claim 25, wherein the controller device is a memory controller.

27. The method of claim 25, wherein the controller device is an I/O controller.

28. The method of claim 20, wherein the system management controller accesses the event data over a system bus of the computer system.

29. The method of claim 28, wherein the system bus is a SMbus.

30. The method of claim 20 further comprising, the system management controller writing the event log in response to accessing the event data.

31. The method of claim 20, wherein writing the event log occurs over a system bus of the computer system.

32. The method of claim 31, wherein the system bus is a SMbus.

EVIDENCE APPENDIX - 37 CFR § 41.37(c)(1)(ix)

None

RELATED PROCEEDINGS APPENDIX - 37 CFR § 41.37(c)(1)(x)

There are no related proceedings.